

**ANL252**

**Python for Data Analytics**

**Tutor-Marked Assignment**

**July 2021 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Fok Jiajun Samuel** | **J2111086** |

**Tutorial Group: ­­­­­­­­­­­­ T09**

**Instructor’s Name: Dr. Munish Kumar**

**Submission Date: 13/08/2021**

# (a)

import math

# (b)

main\_program = True

print(f"Please type in the following data (Press ENTER for default):") # indicating to the user that by pressing Enter

while main\_program: # run the program

print(f"Mean can be any value between minus infinity (–∞) and plus infinity (+∞)")

print(f"Default, mean = 0") # let user know that default mean value is 0

valid\_input = False

while valid\_input == False:# start of control mechanism

mu = input(f"What is the mean: ") # ask for input

if mu == "":# for default settings

mu = float(0) # default value 0

valid\_input = True # end the check

try:

mu = float(mu) # try the value user input if numeric

except ValueError: # value must be numeric

print(f"Your input is not numeric. Please try again. ")

else:

valid\_input = True # end check if pass

continue #continue with the next input

print("Variance must be a value larger than 0")

print(f"Default, variance = 1")

valid\_input = False

while valid\_input == False:# start of control mechanism

variance = input(f"What is the variance: ") # ask for input

if variance == "": # for default settings

variance = float(1) # default value 1

valid\_input = True # end the check

try:

variance = float(variance) # try the value user input if numeric

except ValueError: # value must be numeric

print(f"Your input is not numeric. Please try again. ")

else:

if float(variance) <= 0: # check input value must be larger than 0

print(f"Value must be larger than 0. Please try again ")

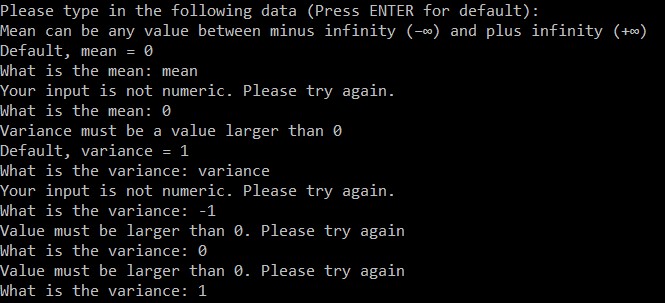
else:

valid\_input = True # end check if pass

continue #continue with the next input

**Picture 1**

Output when user is aske for mean and variance



*Note.* Invalid inputs are made to show control mechanisms

When the user runs the program, The program will indicate to the user to input the values of mean first. If the user presses ENTER, the default value is 0. If the user types a word, the program will ask the user to input a value instead. It will continue until the user inputs a numeric value.

The next input, variance, will then be asked and if the user inputs a word, the program will ask user to input a numeric value. However, if the numeric value is less than or equals to 0 then the program will remind the user to input a value that is greater than 0.

# (c)

print(f"x can be any value between minus infinity (–∞) and plus infinity (+∞)")

valid\_input = False

while valid\_input == False: # start of control mechanism

x = input(f"What is x: ") # ask for input

try:

x = float(x) # try the value user input if numeric

except ValueError: # user have to input something and value must be numeric

print(f"Your input is not numeric or input is empty. Please try again. ")

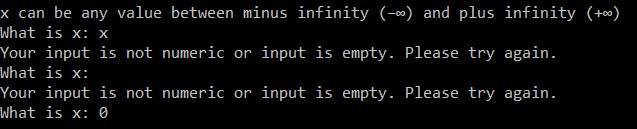
else:

valid\_input = True # end check if pass

main\_program = False # end the loop

**Picture 2**

Output when user is asked value of x



*Note.* Invalid inputs are made to show control mechanisms

The last input by user is the value of x. Since there is no default value of x, the user can input any number. As shown in picture 2, user input ‘x’ or does not input any value, the program will inform the user and ask for another numeric input. The loop ends and the program will calculate the pdf.

# (d)

def pdf(x): #pdf formula

result = ((1 / (math.sqrt(2\*math.pi\*variance))) \* math.exp(-((x - mu)\*\*2) / (2\*variance)))

return result

# (e)

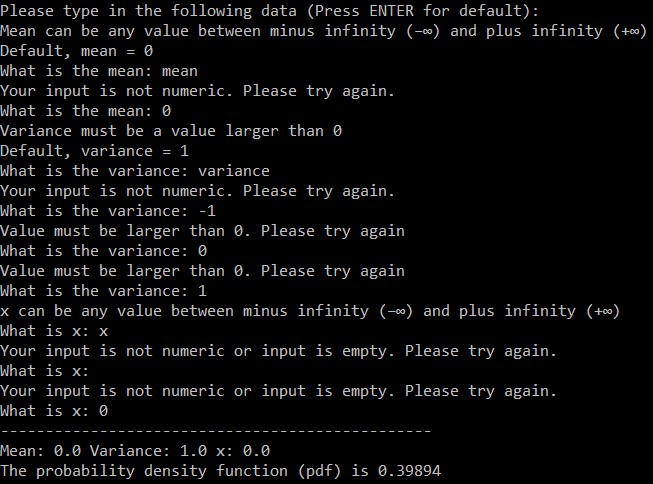
print(f"------------------------------------------------")

print(f"Mean: {mu} Variance: {variance} x: {x}") # informing user of their inputs

print(f"The probability density function (pdf) is {round(pdf(x),5)}") # using the user\_def above to display pdf result

**Picture 3**

Output for pdf result



Note. pdf was calculated using the values that user input in (b) and (c)

Using the inputs by the user in parts (b) and (c), the program calculates the pdf by inputting it in the user defined function which is the pdf formula in part (d)

# (f)

def cdf(x): # alpha is 0.01

list = [] # creating list to store the pdfs

for r in range(0,10001): # for a

a = (x - (r\*0.01)) # formula to find a

list.append(pdf(x = a)) # to add the pdf to the list

cdf = 0.01\*(sum(list)) # cdf formula alpha \* sum of pdfs

return cdf

print(f"(P <= {x}) is {round(cdf(x),5)}") # using the user\_def above and display cdf result

#(f) to check the results for k = 0, 1.64, and 1.96

#user have to input a mean of 0 and variance of 1 at the start, x is set in the list

print(f"------------------------------------------------")

print(f"With Mean = {mu} and Variance = {variance}") # informing user the mean and variance used to calculate the cdf

k = [0,1.64,1.96] # using these values of x to calcualte cdf

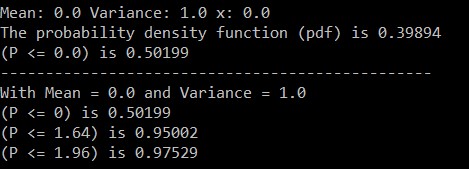
for f in k:

print(f"(P <= {f}) is {round(cdf(x = f),5)}") # using the user\_def above and display cdf result

#(P <= x) is cdf

**Picture 4**

Output of cdf result



*Note.* This is check values of x values (0, 1.64, 1.96)

# (g)

Since, “x = k” from the previous part and the user input the values of mean, variance and x when asked at the beginning of the program. A user defined function (cdf(x)) was created. In the function, a list was created to store the values of the pdf. Since, “a” is a number close to infinity, based off the formula, a = (x-(r\*α)), where “r” is the number or iterations and “α” = 0.01. A “for” loop was used to run through range from 0 to 10,001. After getting “a”, apply “a = x” in the pdf user function to produce the pdf value. As the “for” loop runs, it appends the result to the list. After it reaches the value of “x”, a sum of all the pdf results multiplied by “α” will produce an estimated value for cdf. Since, “α” is set by the programmer, to get a more accurate result, the programmer can change the value to a smaller value. (163 words)

# (h)

dict = {} #creating the dictionary

part\_h = True

h = -5.0 # start value of x to be stored

while part\_h: # start the loop

if h >= 5: # end the loop if x value is 5.0

part\_h = False # end the loop

h = str(h) # converting to string to use as key

dict[h] = round(cdf(x = float(h)),5) # storing the values of cdf in dictionary

h = float(h) # converting back to a float

h += 0.1 # step width of 0.1

h\_1dp= round(h,1) # round key to be x.1 if not it will be x.00000001

h = float(h\_1dp) # setting new h after rounding

print("The probabilities of x between -2.0 and 2.0 with a step width of 0.5:")

print(f"With Mean = {mu} and Variance = {variance}") # informing user the mean and variance used to calculate the cdf

last = True

i = -2.0 # start value of x to display

while last: # start the loop

string\_i = str(i) # converting to string to find key

print(f"{i}:{dict.get(string\_i)}")# key:value

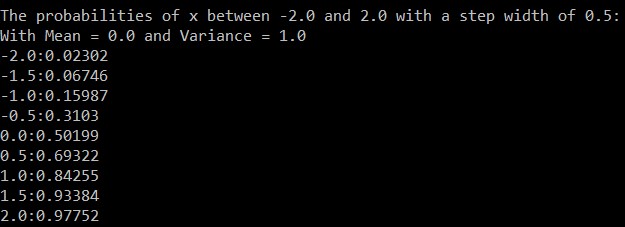
if i == 2.0: # end value of x to display

last = False # end the loop

i += 0.5 # step width of 0.5

**Picture 5**

Output of selected values of dictionary



*Note.* cdf values were calculated with what the user input at the beginning of the program

-5.0 is the start of the dictionary, since it has to be used as a key, it is converted into a string. To store the values into the dictionary, the cdf user defined function was used with “h” replacing “x”. Converting the string back to a float to do the next increment of 0.1 and rounding off to 1 decimal place so that the key could be called for in the next part.

Similar to creating the dictionary, the value “i” is converted into a string so that it can be used to find a key. The program will find all the keys from -2.0 to 2.0 with 0.5 increment.

# **Appendix A**

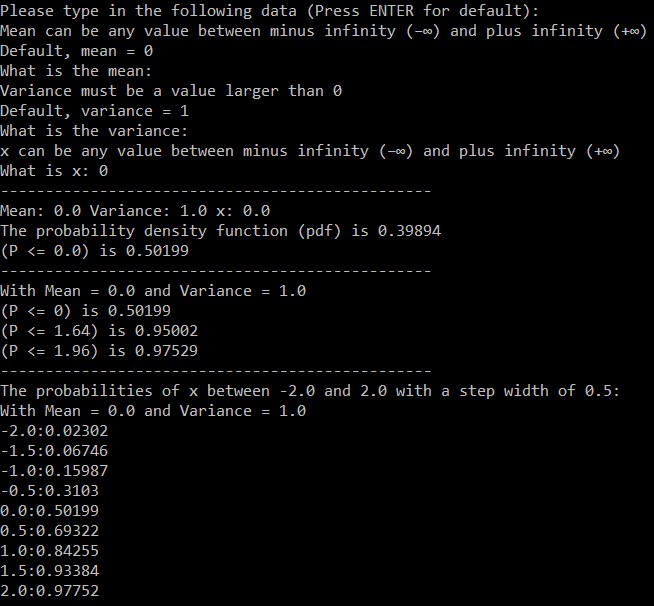
Python program



Copy and paste to desktop, if Microsoft office is blocking it.

# **Appendix B**

# Full output of program



# **Appendix C (copy of code)**

# python ANL252\_TMA01\_samuelfok001\_FokJiajunSamuel.py

# Author: Fok Jiajun Samuel, J2111086

#(a) import math package

import math

#(d) user defined function for pdf

#compute the corresponding probability density fX(x) based on the user inputs in (b) and (c).

def pdf(x): #pdf formula

result = ((1 / (math.sqrt(2\*math.pi\*variance))) \* math.exp(-((x - mu)\*\*2) / (2\*variance)))

return result

# (f) calculating cdf

def cdf(x): # alpha is 0.01

list = [] # creating list to store the pdfs

for r in range(0,10001): # for a

a = (x - (r\*0.01)) # formula to find a

list.append(pdf(x = a)) # to add the pdf to the list

cdf = 0.01\*(sum(list)) # cdf formula alpha \* sum of pdfs

return cdf

#-----------Main Program-------------------------------

#(b) User input for mu and Variance

#mean can be any value between minus infinity (–∞) and plus infinity (+∞)

#variance must be a value larger than 0

#input is numeric

#default is set to mu = 0 and variance = 1

main\_program = True

print(f"Please type in the following data (Press ENTER for default):") # indicating to the user that by pressing Enter

while main\_program: # run the program

print(f"Mean can be any value between minus infinity (–∞) and plus infinity (+∞)")

print(f"Default, mean = 0") # let user know that default mean value is 0

valid\_input = False

while valid\_input == False:# start of control mechanism

mu = input(f"What is the mean: ") # ask for input

if mu == "":# for default settings

mu = float(0) # default value 0

valid\_input = True # end the check

try:

mu = float(mu) # try the value user input if numeric

except ValueError: # value must be numeric

print(f"Your input is not numeric. Please try again. ")

else:

valid\_input = True # end check if pass

continue #continue with the next input

print("Variance must be a value larger than 0")

print(f"Default, variance = 1")

valid\_input = False

while valid\_input == False:# start of control mechanism

variance = input(f"What is the variance: ") # ask for input

if variance == "": # for default settings

variance = float(1) # default value 1

valid\_input = True # end the check

try:

variance = float(variance) # try the value user input if numeric

except ValueError: # value must be numeric

print(f"Your input is not numeric. Please try again. ")

else:

if float(variance) <= 0: # check input value must be larger than 0

print(f"Value must be larger than 0. Please try again ")

else:

valid\_input = True # end check if pass

continue #continue with the next input

#(c) User input for x

#X can be any value between minus infinity (–∞) and plus infinity (+∞)

#input is numeric

print(f"x can be any value between minus infinity (–∞) and plus infinity (+∞)")

valid\_input = False

while valid\_input == False: # start of control mechanism

x = input(f"What is x: ") # ask for input

try:

x = float(x) # try the value user input if numeric

except ValueError: # user have to input something and value must be numeric

print(f"Your input is not numeric or input is empty. Please try again. ")

else:

valid\_input = True # end check if pass

main\_program = False # end the loop

#(e) formatted printing to display the result of (d) to the user.

print(f"------------------------------------------------")

print(f"Mean: {mu} Variance: {variance} x: {x}") # informing user of their inputs

print(f"The probability density function (pdf) is {round(pdf(x),5)}") # using the user\_def above to display pdf result

#(f) to show the CDF result to the user

print(f"(P <= {x}) is {round(cdf(x),5)}") # using the user\_def above and display cdf result

#(f) to check the results for k = 0, 1.64, and 1.96

#user have to input a mean of 0 and variance of 1 at the start, x is set in the list

print(f"------------------------------------------------")

print(f"With Mean = {mu} and Variance = {variance}") # informing user the mean and variance used to calculate the cdf

k = [0,1.64,1.96] # using these values of x to calcualte cdf

for f in k:

print(f"(P <= {f}) is {round(cdf(x = f),5)}") # using the user\_def above and display cdf result

#(P <= x) is cdf

print(f"------------------------------------------------")

#(h) creating the dictionary to store the key and values from -5.0 to 5.0

dict = {} #creating the dictionary

part\_h = True

h = -5.0 # start value of x to be stored

while part\_h: # start the loop

if h >= 5: # end the loop if x value is 5.0

part\_h = False # end the loop

h = str(h) # converting to string to use as key

dict[h] = round(cdf(x = float(h)),5) # storing the values of cdf in dictionary

h = float(h) # converting back to a float

h += 0.1 # step width of 0.1

h\_1dp= round(h,1) # round key to be x.1 if not it will be x.00000001

h = float(h\_1dp) # setting new h after rounding

# remove # to print dict to see what is stored

#print(dict)

print("The probabilities of x between -2.0 and 2.0 with a step width of 0.5:")

print(f"With Mean = {mu} and Variance = {variance}") # informing user the mean and variance used to calculate the cdf

last = True

i = -2.0 # start value of x to display

while last: # start the loop

string\_i = str(i) # converting to string to find key

print(f"{i}:{dict.get(string\_i)}")# key:value

if i == 2.0: # end value of x to display

last = False # end the loop

i += 0.5 # step width of 0.5